Kathleen Sullivan (SBN 242261)
kathleensullivan@quinnemanuel.com
QUINN EMANUEL URQUHART & SULLIVAN LLP
51 Madison Avenue, 22nd Floor
New York, NY 10010
Telephone: (212) 849-7000
Facsimile: (212) 849-7100

Sean S. Pak (SBN 219032)
seanpak@quinnemanuel.com
John M. Neukom (SBN 275887)
johnneukom@quinnemanuel.com.
QUINN EMANUEL URQUHART & SULLIVAN LLP
50 California Street, 22nd Floor
San Francisco, CA 94111
Telephone: (415) 875-6600
Facsimile: (415) 875-6700

Mark Tung (SBN 245782)
marktung@quinnemanuel.com
QUINN EMANUEL URQUHART & SULLIVAN LLP
555 Twin Dolphin Drive, 5th Floor
Redwood Shores, CA 94065
Telephone: (650) 801-5000
Facsimile: (650) 801-5100

Steven Cherny *(admission pro hac vice pending)*
steven.cherny@kirkland.com
KIRKLAND & ELLIS LLP
601 Lexington Avenue
New York, New York 10022
Telephone: (212) 446-4800
Facsimile: (212) 446-4900

Adam R. Alper (SBN 196834)
adam.alper@kirkland.com
KIRKLAND & ELLIS LLP
555 California Street
San Francisco, California 94104
Telephone: (415) 439-1400
Facsimile: (415) 439-1500

Michael W. De Vries (SBN 211001)
michael.devries@kirkland.com
KIRKLAND & ELLIS LLP
333 South Hope Street
Los Angeles, California 90071
Telephone: (213) 680-8400
Facsimile: (213) 680-8500

Attorneys for Plaintiff Cisco Systems, Inc.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA, SAN JOSE DIVISION

CISCO SYSTEMS, INC.,

Plaintiff,

vs.

ARISTA NETWORKS, INC.,

Defendant.

CASE NO. 5:14-cv-5344-BLF

PLAINTIFF CISCO SYSTEMS, INC.'S REPLY CLAIM CONSTRUCTION BRIEF

Tech Tutorial: Mar. 11, 2016
Claim Construction Hearing: Mar. 18, 2016

TABLE OF CONTENTS

I. INTRODUCTION 1

II. ARISTA INCORRECTLY CHARACTERIZES THE ’526 PATENT 1

III. ’526 PATENT CLAIM TERMS 1

A. “management programs” in Claims 1, 10, 14, and 23 1

B. “generic command” in Claims 1, 6, 10, 13, 14, 15, 19, and 23 3

C. “respective command formats,” in Claims 1, 10, 14, 23 5

D. “command parse tree,” in Claims 1, 3, 10, 11, 12, 14-16 5

E. “the validating step including identifying one of the elements as a best match relative to the generic command” in Claims 1 and 14 5

F. “recursively traversing the command parse tree based on an order of the input command words” in Claims 3 and 16 6

G. “the command parse tree having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value” in Claims 1 and 14 6

H. “prescribed command of a selected one of the management programs” in Claims 1 and 14 7

I. “command word translation table, configured for storing for each prescribed command word a corresponding token” in Claims 2 and 15 8

J. “means for validating a generic command received from a user, the validating means configured for specifying valid generic commands relative to a prescribed generic command format and having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value, the validating means identifying one of the elements as a best match relative to the generic command” in Claim 23 8

IV. ARISTA INCORRECTLY CHARACTERIZES THE ’886 PATENT 10

V. ’886 PATENT CLAIM TERMS 10

A. “extensible markup language (XML)” in Claims 1-10 10

B. “command line interface (CLI) parser” / “receiving, with a command line interface (CLI) parser” in Claims 1-5 11

C. “internetwork operating system (IOS) command line interface (CLI) parser subsystem” in Claims 2-4, 7-9 12

D. “XML tag” in Claims 1-10 13

E. “XML parameter” in Claims 1-10 14

F. "parsing the output message to identify at least one CLI token" in Claims 1-10 14

G. "wherein the input command is configured in an extensible markup language (XML) format having a CLI syntax with CLI keywords sequenced according to configuration rules for CLI commands" in Claims 1-10 15

VI. CONCLUSION 15

TABLE OF AUTHORITIES

Page

Cases

Akamai Techs., Inc. v. Limelight Networks, Inc.,
No. 2009-1372, 2015 U.S. App. LEXIS 19848 (Fed. Cir. Nov. 16, 2015)................................15

Atlas IP, LLC v. Medtronic, Inc.,
No. 2015-1071, 2015 WL 6550622 (Fed. Cir. Oct. 29, 2015)..5

Ethicon Endo-Surgery, Inc. v. Covidien, Inc.,
796 F.3d 1312 (Fed. Cir. 2015)..4

Golight, Inc. v. Wal-Mart Stores, Inc.,
355 F.3d 1327 (Fed. Cir. 2004)..9

Hill-Rom Servs., Inc. v. Stryker Corp.,
755 F.3d 1367 (Fed. Cir.) *cert. denied*, 135 S. Ct. 719, 190 L. Ed. 2d 463 (2014) 6

Howmedica Osteonics Corp. v. Wright Med. Tech., Inc.,
540 F.3d 1337 (Fed. Cir. 2008)..7

Markman v. Westview Instruments, Inc.,
52 F.3d 967 (Fed. Cir. 1995)..13

Martek Biosciences Corp. v. Nutrinova, Inc.,
579 F.3d 1363 (Fed. Cir. 2009)..4

Mobile Telecomms. Techs., LLC v. T-Mobile USA, Inc.,
78 F. Supp. 3d 634 (E.D. Tex. 2015) 10, 12

Nystrom v. TREX Co.,
424 F.3d 1136 (Fed. Cir. 2005)..4

Phillips v. AWH Corp.,
415 F.3d 1303 (Fed. Cir. 2005)..12

Teleflex, Inc. v. Ficosa N. Am. Corp.,
299 F.3d 1313 (Fed. Cir. 2002)..9

Walker Digital, LLC v. Microsoft Corp.,
590 F. App’x. 956 (Fed. Cir. 2014)...14

I. INTRODUCTION

Arista's proposed constructions for the '526 Patent are based on an incomplete and misleading treatment of the inventions claimed in that patent. The '526 Patent describes an inventive management system and architecture that allow the user to enter generic commands in order to execute commands on separate management programs. Although the inventions include a specialized parser for interpreting generic commands and a set of translators for issuing specific commands to management programs, it cannot be reduced simplistically to a translator of commands, as Arista's flawed constructions suggest.

Arista's overly narrow proposed constructions for the '886 Patent improperly cherry-pick from the intrinsic and extrinsic record, while simply ignoring the voluminous evidence that contradicts those constructions. Cisco's proposed constructions, on the other hand, are supported by and consistent with ***all*** of the intrinsic and extrinsic evidence.

II. ARISTA INCORRECTLY CHARACTERIZES THE '526 PATENT

Purporting to quote the '526 Patent, Arista asserts that the "prescribed commands" in the '526 Patent are "unique to each management program." (Opp. at 2 (citing '526, Ex .1[1] at 1:48-63).[2] However, the '526 specification passage that Arista cites never uses the word "unique" to describe the prescribed commands. In fact, the '526 specification as a whole never uses the word "unique" to describe the prescribed commands.

Arista also misleadingly points to Figure 2 as somehow defining the "command parse tree," but this figure does not and cannot define the "command parse tree." Figure 2 only describes an embodiment and does not limit the full scope of the invention. The '526 specification makes clear that Figure 2 is "illustrating in detail the parser of FIG. 1 ***according to an embodiment*** of the present invention." ('526 at 2:48-49 (emphasis added)).

III. '526 PATENT CLAIM TERMS

A. "management programs" in Claims 1, 10, 14, and 23

Cisco's Construction	Arista's Construction

[1] Hereinafter "'526."
[2] "Opp." refers to "Arista's Responsive Claim Construction Brief" (Dkt. 142).

"separate tools or external agents having their own respective command formats that provide management functions"	"tools that are configured to execute user-entered commands having their own respective command formats rather than the generic command format"

First, Arista asserts that management programs must be configured to accept only "user-entered commands," arguing that the invention "would not achieve any particular benefit" if it encompassed systems using automated machine language instructions. (Opp. at 3). In other words, Arista contends that the '526 Patent cannot use automated machine language instructions because if it did, the user would be not be saved from having to learn particular commands (since those functions would be carried out by machine instructions, not user-entered commands). This argument, however, is based on a misunderstanding of the invention of the '526 Patent. As set forth in the '526 specification and confirmed by the testimony of the inventor J. Wheeler, the invention was not simply a translator. ('526 at 1:7-27). Rather, the invention automates the configuration process, which previously had been largely manual and therefore prone to error. (*See* Wheeler Depo. Tr. (November 12, 2015), Ex. 9[3] at 71:12-72:24; 80:24-81:15). The '526 invention was developed to automate the multi-step process of setting up and administering a device. No distinction is made in the intrinsic evidence between user-entered commands and other instructions that are carried out by the machine. ('526 at 1:10-2:37).

Second, Arista argues that if the "generic commands" simply duplicated the original commands from the management programs, the purpose of the invention would be frustrated. But again, this argument is premised on a misunderstanding of the '526 invention as simply a translator. Contrary to Arista's argument, the primary purpose of the invention as an automated configuration or management system can be achieved even if there is some overlap between the "generic commands" and the commands from the management programs. ('526 at 1:7-27). In addition, nothing in Arista's argument actually ***requires*** that there be absolutely no overlap at all between "generic commands" and "respective command formats." If a particular command for a management program is easy to remember, reasonably short and fits within the hierarchy and organization of the other generic commands, nothing in the '526 Patent prohibits using that as a

[3] Hereinafter, "Wheeler Tr." Ex. 9 refers to Exhibit 9 of the Tung Declaration.

generic command. Moreover, nothing in the specification or claims requires introducing an artificial prohibition against overlap between the generic commands and the respective command formats that Arista proposes through the "rather than" in its construction.

Finally, Arista argues that the passage quoted in Cisco's opening brief ('526 at 3:1-15) does not support Cisco's construction. First, Cisco's construction properly describes "management programs" as "separate" ***or*** "external." The use of "or" in Cisco's construction makes clear there is no absolute requirement that the "management programs" must be external. Not only is this clear in the cited specification passage describing Figure 1, it is further supported by Figure 1 itself, which shows management programs 18a and 18b within the system 10, while 18c and 18d are outside the system 10. At the same time, if there were no requirement that "management programs" be "separate" or "external," then the management programs that the system is supposed to execute as part of its operation and administration management (OAM) functionality could possibly be within the system itself. But running programs within the system itself is not managing those programs; it is simply running the system itself, which does not accurately reflect the '526 invention. Regarding the question "separate from what?" that Arista raised, Professor Almeroth testified at his deposition that "separate" refers to separate routines or programs that are separate from each other, or separate processes, in contrast to referring to physically separate or external computers. (*See* Almeroth Depo. Tr. (November 23, 2015), Ex. 10[4] at 101:24-107:7).

B. "generic command" in Claims 1, 6, 10, 13, 14, 15, 19, and 23

Cisco's Construction	Arista's Construction
"command that provides an abstraction of the tool-specific command formats and syntax, enabling a user to issue the command based on the relative functions, as opposed to the specific syntax for a corresponding tool"	"command having a format and syntax that is an abstraction of the command formats and syntaxes of more than one management program, as opposed to the specific syntax for any such management program"

Arista misleadingly argues that "generic command" is indefinite based on testimony from inventor J. Wheeler and Prof. Almeroth that they do not understand this term ***outside the context***

[4] Hereinafter, "Almeroth Tr." Ex. 10 refers to Exhibit 10 of the Tung Declaration.

of the patent. (*See, e.g.*, Almeroth Tr. at 128:7-10 (explaining answer as "outside the confines of this … patent")). But, that is not the correct standard for indefiniteness. It is perfectly appropriate, and common, for a patentee to provide context for understanding of a claim term in the patent specification itself. *Nystrom v. TREX Co.*, 424 F.3d 1136, 1143 (Fed. Cir. 2005). Arista implies that Prof. Almeroth's proper consultation of the patent to interpret the meaning of "generic command" somehow indicates that the term is indefinite. This misapplies the law. It is black letter law that a claim term is to be understood by one of ordinary skill ***in the context of the patent***. *See Ethicon Endo-Surgery, Inc. v. Covidien, Inc.*, 796 F.3d 1312, 1338 (Fed. Cir. 2015) (considering whether "one of ordinary skill in the art, ***in view of the specification***, would understand the scope of the claims … with reasonable certainty") (emphasis added).

Furthermore, Arista misinterprets Prof. Almeroth's deposition testimony describing why he could not answer Arista's questions on generic commands. Prof. Almeroth consistently and repeatedly explained the need for proper context (which was not provided by the questioning attorney) in order to answer questions about whether certain isolated commands were generic commands, since the key test was whether a generic command represented an abstraction. (*See, e.g.,* Almeroth Tr. at 127:9-138:20). For example, Prof. Almeroth testified that, if a particular command were written down by itself on a piece of paper, in isolation, the answer to the question would be: "It depends." (*Id*. at 128:4-12). The need for context to determine if a command represents an abstraction of "tool-specific commands and formats," and thus whether it is a generic command or not, does not render the term "generic command" indefinite.

In fact, the term "generic command" is described very specifically in the '526 specification and Cisco's construction is based ***verbatim*** on that disclosure without modification. (*See* '526 Patent at 3:31-35). When a patent term has been explicitly defined in the specification, as "generic command" has been here, that definition trumps any extrinsic testimony. *Martek Biosciences Corp. v. Nutrinova, Inc.*, 579 F.3d 1363, 1382 (Fed. Cir. 2009). Thus, in light of the '526 specification's explicit and clear description of "generic command," the claim term "generic command" distinctly sets forth the claim scope with particularity.

Arista never responds to the reasons Cisco provided in its opening brief that Arista's construction is also improper because of the limitation "more than one management program."

C. "respective command formats," in Claims 1, 10, 14, 23

Cisco's Construction	Arista's Construction
"command format specific to a management program"	command names and syntaxes specific to each management program

Arista does not make any substantive arguments in its brief with respect to this term, and Cisco refers to its arguments in its opening brief on this term.

D. "command parse tree," in Claims 1, 3, 10, 11, 12, 14-16

Cisco's Construction	Arista's Construction
"a hierarchical data representation having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value"	"tree": "data structure consisting of linked nodes, with a root node (a node with no parent nodes), and where the remaining nodes are either a branch node (a node with a parent node and one or more children nodes), or a leaf node (a node with a parent node and no children nodes)" "command parse tree": "tree for interpreting commands where each node corresponds to a command component"

Arista's construction, focused on the phrase "data structure" from extrinsic definitions of "tree," misses the point of what a tree is. Arista's construction is directed to how a tree is stored, but a tree is defined by the relationships between its nodes. Just as a letter is not defined by whether or not it is put into an envelope and mailed, a tree is not defined by how it is stored.

Arista criticizes Cisco's use of claim language in its proposed construction, but there is nothing wrong with Cisco's reference to claim language in its proposed construction to reflect the plain meaning of "command parse tree" as used in the claim. *Atlas IP, LLC v. Medtronic, Inc.*, No. 2015-1071, 2015 WL 6550622, at *4 (Fed. Cir. Oct. 29, 2015).

E. "the validating step including identifying one of the elements as a best match relative to the generic command" in Claims 1 and 14

Cisco's Construction	Arista's Construction
Plain and ordinary meaning (except for specific terms appearing within the phrase that are otherwise terms for construction)	Indefinite, OR if not indefinite: "the validating step having the capability of both identifying the element in the parse tree that exactly matches the generic command, and, in the absence of an exact match, identifying the element that contains the last validated component of the generic command"

Arista argues that the disclosed embodiments can be described as an "exact match" or "absence of an exact match," and has proposed a construction with those terms. But neither of the descriptions in the '526 Patent that Arista relies on actually uses the term "exact match."

More importantly, nothing in the ’526 Patent indicates that the two embodiments relied on by Arista are the only possible examples of “best match.” Rather, these two embodiments indicate that what Arista calls an “exact match” is ***one*** possible way to identify a “best match,” but a person of ordinary skill would recognize that a parser can and does use other techniques to identify a best match. For example, Wheeler testified that the system he developed had auto-complete functionality. (*See* Wheeler Tr. at 89:15-90:1; 90:19-92:22). According to Wheeler, the parser would recognize even incomplete words, and would not look at only the last valid word, as Arista’s construction requires. Arista’s proposal is improperly limited to specific embodiments and fails to capture the full scope of the term as understood by one of ordinary skill in the art.

F. “recursively traversing the command parse tree based on an order of the input command words” in Claims 3 and 16

Cisco’s Construction	Arista’s Construction
Plain and ordinary meaning (except for specific terms appearing within the phrase that are otherwise terms for construction)	“recursively”: “by using a function that calls itself,” “traversing the command parse tree based on an order of the input command words”: “sequentially determining the presence of each word of an input command in a node of a command parse tree, such that the order of the words (e.g., first, second, third) corresponds to the hierarchy of the nodes (e.g., parent, child, grandchild).”

Arista states this term needs to be construed “because jurors will not know what ‘recursive’ means.” This is not the correct standard – terms need to be understood by one of ordinary skill in the art in light of the specification, not by laymen who do not have ordinary skill in the art. *Hill-Rom Servs., Inc. v. Stryker Corp.*, 755 F.3d 1367, 1371 (Fed. Cir.) *cert. denied*, 135 S. Ct. 719, 190 L. Ed. 2d 463 (2014).

Arista points to four dictionary definitions of “recursive,” but does not address the fact that all four describe only recursive functions or routines, not recursive traversal of a tree, which is the term at issue. Instead, Cisco is the only party who has provided evidence of the plain and ordinary meaning of “recursively traversing,” as set forth in Almeroth Decl. at ¶¶ 71-80.

G. “the command parse tree having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value” in Claims 1 and 14

Cisco’s Construction	Arista’s Construction
Plain and ordinary meaning (except for	“elements”: “nodes” “command action value”: “piece of data that uniquely represents the

specific terms appearing within the phrase that are otherwise terms for construction)	prescribed command.” the entire phrase: “the command parse tree having nodes, such that each node specifies a unique command action value for each generic command component.”

Arista argues this term requires a “one-to-one pairing between ‘generic command components’ and ‘command action values,’” but the plain language of the term makes clear this is not so. The relevant portion of this term “corresponding at least one command action value” includes the language “at least one” which means the relationship between generic command component and command action value is not necessarily a one-to-one correspondence.

Arista misinterprets the cited testimony from the ’526 inventor J. Wheeler when it concludes that Appendix A is “useless when trying to glean how generic commands map to prescribed commands.” (Opp. at 12). In the cited passage (Wheeler Tr. at 238:1-241:16 (Dkts. 141, 150)), the inventor explains that he does not understand “generic command” ***outside the context*** of the patent, and confirms that the headings “new syntax” and “old command line/syntax” in Appendix A are appropriate. He does not comment on how and whether generic commands map to prescribed commands. In the other cited passage (*id*. at 233:1-13), the inventor explains that certain developer tools were formerly internal tools only, which has nothing to do with Arista’s allegation about mapping of generic and prescribed commands. Regardless, inventor testimony “cannot be relied on to change the meaning of the claims.” *Howmedica Osteonics Corp. v. Wright Med. Tech., Inc.*, 540 F.3d 1337, 1346 (Fed. Cir. 2008). And as with the “recursively traversing” term, Arista again argues that “jurors will not understand it” as a reason to construe a term. This is the incorrect standard as described above in III.F.

H. “prescribed command of a selected one of the management programs” in Claims 1 and 14

Cisco’s Construction	Arista’s Construction
Plain and ordinary meaning (except for specific terms appearing within the phrase that are otherwise terms for construction)	“one of the commands in the respective command format of a specific management program”

Arista points to the phrase “selected one of the management programs” as support for its construction requiring a specific management program. But Arista’s supporting logic is flawed: the phrase “selected one of” implies there must be more than one to select from. If Arista’s only intention with this construction is that the prescribed command is in a format that at least one management program can understand, Cisco does not disagree. But there is nothing limiting the prescribed format to be unique to one and only one specific management program.

I. "command word translation table, configured for storing for each prescribed command word a corresponding token" in Claims 2 and 15

Cisco's Construction	Arista's Construction
Plain and ordinary meaning (except for specific terms appearing within the phrase that are otherwise terms for construction)	"command word translation table": "data structure with rows and columns that translates a command word into a token" "prescribed command word": "valid generic command word." "token": "letter, number, ~~or~~ symbol, [**<u>or other short code</u>**] that either (1) uniquely represents a prescribed command word, or (2) indicates the presence of an invalid command word" The remainder of this phrase does not require construction.

Arista claims that Cisco's description of the plain meaning of a translation table (as a correspondence between a prescribed word and a token) reads out the meaning of the word "table" from the claim. Arista is wrong. It does precisely the opposite. The key idea behind a translation table is the matching, or correspondence, between two entities. Nothing in the '526 Patent suggests that it is necessary to store the correspondence as a "data structure with rows and columns" or that this is the only way to store this information. Other types of data structures can implement the translation table while carrying out the purpose of the invention.

Arista proposes modifying their construction of "token" to include "or other short code," so that it is no longer limited to only a "letter, number, or symbol." With this change, the parties no longer have a dispute over the meaning of token.

J. "means for validating a generic command received from a user, the validating means configured for specifying valid generic commands relative to a prescribed generic command format and having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value, the validating means identifying one of the elements as a best match relative to the generic command" in Claim 23

Cisco's Construction	Arista's Construction
For "means for validating a generic command received from a user": <u>Function</u>: validating a generic command received from a user. <u>Structure</u>: Parser 14 in Figure 2,	<u>Functions:</u> (1) validating a generic command received from a user (2) specifying valid generic commands relative to a prescribed generic command format, (3) having elements each specifying at least one corresponding

which includes the command word translation table 20 and the command parse tree 22, as described in 3:36- 61, and equivalents For rest of term, plain and ordinary meaning (except that specific terms appearing within the phrase should be construed as proposed above)	generic component and a corresponding at least one command action value, and (4) identifying one of the elements as a best match relative to the generic command. Disclosed structure: A processor executing a parser, and a corresponding memory storing a command parse tree, wherein the parser executes the algorithm of Figure 3, and wherein (1) each node of the command parse tree specifies one token and a corresponding command key; (2) the top-level nodes of the command parse tree represent all possible valid first words in the input command, second-level nodes represent all possible valid second words for each valid first word in the input command, and so on;

The crux of the parties' dispute concerns the identification of the structure for the functions or sub-functions. Arista incorrectly characterizes Cisco's argument as identifying a structure only able to handle valid commands, and implies that the structure cannot determine a best match. To the contrary, the structure that Cisco identified is specifically described in the intrinsic evidence as determining a best match. Indeed, the relevant portion of the '526 specification reads: "a command parse tree 22…[a] is configured for validating a received generic command…to determine…a tree element 24 identified as ***a best match***." ('526 at 3:36-51(emphasis added)).

Moreover, what Arista argues are missing from Cisco's structures are not structures at all but in fact functional elements. Arista criticizes Cisco's structures as incomplete because they do not include Figure 3 of the '526 Patent and the corresponding description in the specification (3:62-4:54), and therefore do not "explain[] how Cisco's cited embodiment works." (Opp. at 16). But Figure 3 is a flow chart, which Arista describes as the "decisional logic that applies to these data structures," not the data structures themselves (which Cisco's construction already identified). (*Id.*). Arista is attempting to improperly limit the scope of the means-plus-function claim (beyond the function disclosed in the claims and the structure disclosed in the specification) to one method disclosed in an embodiment in the patent. That is contrary to the law. *Golight, Inc. v. Wal-Mart Stores, Inc.*, 355 F.3d 1327, 1335 (Fed. Cir. 2004) (finding no reason to import the preferred embodiment of a means plus function claim as a limitation during claim construction); *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1326 (Fed. Cir. 2002).

IV. ARISTA INCORRECTLY CHARACTERIZES THE ’886 PATENT

To support its erroneous constructions, Arista provides a “factual overview” of the ’886 Patent that mischaracterizes the ’886 Patent in several ways. First, Arista misleadingly states “[t]he ’886 Patent sought to provide an easier, more structured approach to sending input to—and receiving output from—Cisco routers.” (Opp. at 16). However, the ’886 Patent is not limited to interfacing with Cisco routers. Arista cites to ’886 Patent at 1:26-38 to support its statement, but column 1:26-38 is in the “BACKGROUND” and “Related Art” sections of the ’886 Patent, which do not limit the claims. *See Mobile Telecomms. Techs., LLC v. T-Mobile USA, Inc.*, 78 F. Supp. 3d 634, 652-53 (E.D. Tex. 2015). Thus, Arista’s assertion that the ’886 Patent is limited to Cisco routers is incorrect. Second, Arista asserts without support that “[i]n all of the embodiments, as well as all of the claims, th[e] structured format [for input commands] is XML.” (Opp. at 16). As explained immediately below, claims of the ’886 Patent are not limited to a particular XML standard.

V. ’886 PATENT CLAIM TERMS

A. “extensible markup language (XML)” in Claims 1-10

Cisco’s Construction	Arista’s Construction
“extensible”: a property of a computer language that allows the user to add new features or modify existing ones “markup language”: a computer language that allows the user to add identifiers to a document for indicating logical components or layout	markup language defined by one of the versions of the XML standard published by the W3C organization

The crux of the dispute on this term is whether the Court should construe “extensible markup language (XML)” according to its plain and ordinary meaning, as Cisco proposes, or if it should be improperly limited to a particular embodiment, as Arista proposes. The entirety of Arista’s argument with respect to this term suffers from a simple but fatal flaw: Arista focuses on the initials “XML” and its use in the specification, but completely ignores the claim term itself.

As Cisco’s expert Dr. Almeroth explained, the meanings of “extensible markup language” and “XML,” as they are commonly understood in the art, do not completely overlap. The term “extensible markup language” is a descriptive term that does not denote to one of ordinary skill in the art a particular language; whereas, “XML” may refer to a particular example of an extensible markup language called the "eXtensible Markup Language." (Almeroth Decl. at ¶¶ 39-41; *see also* Almeroth Tr. at 25:14-26:6).[5] Thus, the term "extensible markup language," in lower case, is a broader term than the capitalized "eXtensible Markup Language." Capitonyms, such as "Earth" meaning the planet versus "earth" meaning soil, are common in the English language. In an attempt to improperly limit the claims, Arista argues that "XML" in the claims refers to the capitalized "eXtensible Markup Language." (Opp., Exs. 6-8). But Arista completely ignores that the broader lower-case "extensible markup language" term is already explicitly recited in the claims. In the context of the claim term "extensible markup language (XML)," the initials "XML" refer to the lower-case term as a shorthand abbreviation and are not limited to a particular standard as Arista suggests. And even though the specification of the '886 uses "XML" to mean "eXtensible Markup Language," as explained in Cisco's opening brief, the specification also repeatedly states the invention is not limited to "eXtensible Markup Language." (Opening at 18[6]; *see also* '886, Ex. 2 at 3:26-29; 3:50-52; 4:27-30). Accordingly, Arista's attempt to limit the claims should be rejected because it is contrary to the plain and ordinary meaning of "extensible markup language" and contrary to the explicit teachings of the intrinsic evidence.

Arista also set forth analogies in arguing its position. But Arista's analogies only further confuse the issue. For example, Arista analogizes "extensible markup language (XML)" to "Uniform Resource Locator (URL)." (Opp. at 18). However, the terms are not analogous. "Uniform Resource Locator" is a specific term of art defined in a single document,[7] but there is no such specific definition for "extensible markup language." As Dr. Almeroth explained, "extensible markup language" is a descriptive term and encompasses different languages such as XHTML, SOAP, and eXtensible Markup Language. (Almeroth Decl. at ¶¶ 39-43).

B. "command line interface (CLI) parser" / "receiving, with a command line interface (CLI) parser" in Claims 1-5

Cisco's Construction	Arista's Construction

[5] Arista's unsupported assertion that Dr. Almeroth's opinions are *ipse dixit* is factually wrong. Dr. Almeroth's declaration cited to numerous pieces of intrinsic and extrinsic evidence to support his opinions. (*See* Almeroth Decl. at ¶¶ 39-47).

[6] "Opening" refers to Cisco's Opening Claim Construction Brief (Dkt. 91).

[7] *See* IETF RFC 1738 (1994), *available at* https://www.ietf.org/rfc/rfc1738.txt.

Claim term: "command line interface (CLI) parser" Proposed construction: a component of the routing system for analyzing command line interface (CLI) commands using a grammar	Claim term: "receiving, with a command line interface (CLI) parser" Proposed construction: "receiving": taking as an input "a command line interface (CLI) parser": a program that breaks down the individual sub-parts of a command using a CLI grammar

Arista improperly narrowly construes the claim term "parser" as a program that only breaks down input. Cisco used both intrinsic and extrinsic evidence to show that parsing ***may*** involve breaking down the input into sub-components, but that is not required. (Opening at 19-20). Arista, in contrast, relied only on extrinsic evidence. Because Cisco's construction comports with the intrinsic and extrinsic evidence, but Arista only cherry-picks extrinsic evidence to support its construction, Cisco's construction should be adopted. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1317 (Fed. Cir. 2005) ("[E]xtrinsic evidence…is 'less significant than the intrinsic record.'") (citation omitted). Arista also argues that Cisco's construction merely replaces "parsing" with "analyzing." This is not the case. Rather, Cisco's construction explains that "parsing" is "analyzing…using a grammar," which comports with the intrinsic and extrinsic evidence. (Opening at 18-20).

C. "internetwork operating system (IOS) command line interface (CLI) parser subsystem" in Claims 2-4, 7-9

Cisco's Construction	Arista's Construction
a subcomponent of the internetwork operating system (IOS) for analyzing command line interface (CLI) commands using a grammar.	"internetwork operating system (IOS) command line interface (CLI)": the CLI interface designated by Cisco as IOS CLI and that is used for configuring, monitoring, and maintaining Cisco devices. Also referred to as 'IOS/CLI.' the entire phrase: processor or portion thereof that executes a program that breaks down the individual sub-parts of a command using the grammar of IOS/CLI

While Arista purportedly cites to intrinsic and extrinsic evidence to support its contention that this claim term should be limited to Cisco's IOS product, the alleged evidence falls short. Arista cites to the Background and Related Art sections of the '886 Patent (Opp. at 21), but these sections do not limit the claims. *See, e.g., Mobile Telecomms.*, 78 F. Supp. 3d at 652-53 (rejecting a proposed construction that would limit a claim term to "Descriptions of Related Art"). Next, Arista contends that the inventors agree that "IOS" refers to Cisco's product. (Opp. at 20).

However, the lines of questioning cited by Arista were not in the context of the ’886 Patent, but rather specifically in the context of the inventors’ work at Cisco or with Cisco’s products. (Opp., Ex. 1 at 86:4-19; Ex. 9 at 21:1-11). Thus, it is not surprising that the inventors, when discussing their work at Cisco, would characterize “IOS” as Cisco’s product. But that does not limit the claims. *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 983 (Fed. Cir. 1995) (1996) (holding that testimony of an inventor “cannot be relied on to change the meaning of the claims”).

As explained in Cisco’s opening brief, Arista’s proposed construction creates the absurd result that the only entity that can possibly infringe a claim in a Cisco-owned patent is Cisco itself. Arista’s response that “[a]n entity other than Cisco could potentially use, make, or sell a subsystem that interfaced with Cisco’s IOS CLI product” fares no better. (Opp. at 21). If the claims are construed as Arista proposes, then Cisco obtained these claims so as to assert them only against Cisco itself or Cisco’s users and customers, rather than Cisco’s competitors that copy its technology, like Arista. This is clearly incorrect.

D. “XML tag” in Claims 1-10

Cisco’s Construction	Arista’s Construction
one or a pair of XML indicators identifying data “extensible markup language” or “XML” as construed above	indicator that marks the start or end of an XML element and that is designated by a starting angle bracket, a corresponding closing angle bracket, and the content in between

Arista mischaracterizes Cisco’s proposed construction as an attempt to expand beyond the scope of the term’s standard technical meaning. (Opp. at 23). This is incorrect. Cisco’s construction is consistent with the plain and ordinary meaning of the term. The parties agree that a tag must include at least one bracketed indicator that has some text surrounded by two brackets. But the crux of the dispute is whether, as Cisco proposes, “tag” in the singular can also refer to a pair of such bracketed indicators. As explained in Cisco’s opening brief, “tag” indeed may refer to a pair of identifiers. For example, the extrinsic evidence cited by Arista itself states: “[u]sually ***one pair*** of angle brackets is placed before an element, and another pair is placed after, to indicate where the element begins and ends.” (Opp., Ex. 7 at 511 (emphasis added)). The purpose of a tag is to identify a portion or an element in a document. To do that, it must be able to indicate the beginning and the end of the element. Therefore, a tag almost always comes in a pair of bracketed indicators, one to indicate the beginning and one to indicate the end. As such, the pair of indicators is commonly understood in the art as a single tag, because both are necessary to identify a portion of a document.

As explained in Cisco's opening brief, this is also consistent with the intrinsic evidence, where the specification uses the term "tag" to refer to both a single or a pair of bracketed XML indicators. (Opening at 22). Arista complains that the '886 specification at 5:49-52 is unclear. (Opp. at 23). Arista is wrong. As explained in Cisco's opening brief, that portion of the specification, when read in context and in conjunction with Table 1, clearly indicates that "tag" in the singular can refer to both a single or a pair of XML indicators. (Opening at 22). Furthermore, as explained above, Cisco's proposed construction is consistent with the plain meaning of "tag." Therefore, the portion of the specification cited by Cisco is not a disclaimer, nor is Cisco arguing that the patentees are acting as their own lexicographers for the term "tag." Accordingly, contrary to Arista's arguments, the intrinsic evidence cited by Cisco does not need to meet the "clear and unmistakable" standard. *Walker Digital, LLC v. Microsoft Corp.*, 590 F. App'x. 956 (Fed. Cir. 2014) (holding the claim term "auction" is not limited to only one sale in part because the term, as used in the specification, encompassed multiple sales).

E. "XML parameter" in Claims 1-10

Cisco's Construction	Arista's Construction
Plain and ordinary meaning, except for "XML" as construed above	indicator within an XML tag that signals that the tag includes a CLI keyword

Arista made no arguments to support its construction. (Opp. at 23). This only further indicates that Arista's rewording of this claim term is unnecessary.

F. "parsing the output message to identify at least one CLI token" in Claims 1-10

Cisco's Construction	Arista's Construction
analyzing the output message to extract at least one unit of CLI characters in a sequence	breaking down the output message into its constituent character strings and determining that at least one such string corresponds to a CLI keyword or parameter

Arista presents two arguments in support of its construction. First, Arista repeats its argument that "parsing" should mean "breaking down." But for all the reasons set forth in Section V.B above, Arista's construction should be rejected. Next, Arista wrongly contends that its construction aligns with the specification. To the contrary, as explained in Cisco's opening brief, Arista's construction actually reads out the preferred embodiment. (Opening at 24). Notably, Arista provides no response to Cisco's arguments. (Opp. at 23-24).

G. "wherein the input command is configured in an extensible markup language (XML) format having a CLI syntax with CLI keywords sequenced according to configuration rules for CLI commands" in Claims 1-10

Cisco's Construction	Arista's Construction
Plain and ordinary meaning, except for "extensible markup language (XML) format" as construed above	"an extensible markup language (XML) format": a format that complies with one of the versions of the XML standard published by the W3C organization "keyword": word describing an action or operation that the computer can recognize and execute The entire phrase: wherein the input command is written in an extensible markup language (XML) format, such that one or more CLI keywords, on the one hand, and the CLI parameters, on the other, are contained within respective XML tags, and the sequence of the tags complies with the sequencing rules for keywords and parameters in the CLI grammar and syntax

Arista ostensibly offers its construction to provide "guidance" for the term, but in actuality, its construction is an attempt to narrow the term to a particular embodiment disclosed in the specification. (Opening at 25). Arista attempts to justify its construction by arguing that "the disclosure contained [in the specification] is *the only description of the claim phrase in the entire specification*." (Opp. at 25 (emphasis in original)). But that position is contrary to well-established Federal Circuit law. "'[E]ven where a patent describes only a single embodiment, claims will not be read restrictively unless the patentee has demonstrated a clear intention to limit the claim scope using words or expressions of manifest exclusion or restriction.'" *Akamai Techs., Inc. v. Limelight Networks, Inc.*, No. 2009-1372, 2015 U.S. App. LEXIS 19848, *14 (Fed. Cir. Nov. 16, 2015) (citation omitted). There are no expressions of manifest exclusion anywhere in the intrinsic evidence. To the contrary, the '886 Patent explicitly states that the invention is not bounded by the disclosed embodiments. (*See, e.g.,* '886 , Ex. 2 at 7:22-34). Arista's argument that the claims should be construed to one embodiment because it is the only disclosed embodiment is unavailing.

VI. CONCLUSION

For all the reasons stated above, Cisco's proposed constructions should be adopted and Arista's proposed constructions should be rejected.

DATED: December 21, 2015

Respectfully submitted,

/s/ Sean S. Pak

Kathleen Sullivan (SBN 242261)
kathleensullivan@quinnemanuel.com
QUINN EMANUEL URQUHART &
SULLIVAN LLP
51 Madison Avenue, 22nd Floor
New York, NY 10010
Telephone: (212) 849-7000
Facsimile: (212) 849-7100

Sean S. Pak (SBN 219032)
seanpak@quinnemanuel.com
John M. Neukom (SBN 275887)
johnneukom@quinnemanuel.com.
QUINN EMANUEL URQUHART &
SULLIVAN LLP
50 California Street, 22nd Floor
San Francisco, CA 94111
Telephone: (415) 875-6600
Facsimile: (415) 875-6700

Mark Tung (SBN 245782)
marktung@quinnemanuel.com
QUINN EMANUEL URQUHART &
SULLIVAN LLP
555 Twin Dolphin Drive, 5th Floor
Redwood Shores, CA 94065
Telephone: (650) 801-5000
Facsimile: (650) 801-5100

Steven Cherny (admitted *pro hac vice*)
steven.cherny@kirkland.com
KIRKLAND & ELLIS LLP
601 Lexington Avenue
New York, New York 10022
Telephone: (212) 446-4800
Facsimile: (212) 446-4900

Adam R. Alper (SBN 196834)
adam.alper@kirkland.com
KIRKLAND & ELLIS LLP
555 California Street
San Francisco, California 94104
Telephone: (415) 439-1400
Facsimile: (415) 439-1500

Michael W. De Vries (SBN 211001)
michael.devries@kirkland.com
KIRKLAND & ELLIS LLP

333 South Hope Street
Los Angeles, California 90071
Telephone: (213) 680-8400
Facsimile: (213) 680-8500

Attorneys for Plaintiff Cisco Systems, Inc.

PROOF OF SERVICE

I, Jason L. Liu, declare that I am over the age of eighteen (18) and not a party to the entitled action. I am an attorney in the law firm of Quinn Emanuel Urquhart & Sullivan LLP, and my office is located at 50 California Street, 22nd Floor, San Francisco, CA 94111. On December 21, 2015, I caused to be filed the following:

PLAINTIFF CISCO SYSTEMS, INC.'S REPLY CLAIM CONSTRUCTION BRIEF

with the Clerk of the Court using the Official Court Electronic Document Filing System, which served copies on all interested parties registered for electronic filing.

I declare under penalty of perjury that the foregoing is true and correct. Executed on December 21, 2015, at San Francisco, California.

/s/ Jason L. Liu
Jason L. Liu